JSC-14790
APR 5 1979
CR160161

Lockheed Electronics Company, Inc.
A SUBSIDIARY OF LOCKHEED CORPORATION
1830 NASA Road 1, Houston, Texas 77058
Tel. 713-333-5411

Ref: 642-6963
NAS 9-15800
Job Order 73-743-20

TECHNICAL MEMORANDUM

USER'S GUIDE FOR THE CATE-LIEBIG LAW OF THE MINIMUM

By

D. E. Phinney and R. B. Cate

(NASA-CR-160161) USER'S GUIDE FOR THE CATE-LIEBIG LAW OF THE MINIMUM (Lockheed Electronics Co.) 53 p HC A04/MF A01 CSCL 09B

N79-21801

Unclas
G3/61 24525

Approved By: B L Carroll
B. L. Carroll, Manager
EO Development and Evaluation
Department

February 1979

LEC-12377

CONTENTS

Section Page

1. GENERAL DESCRIPTION OF THE COMPUTER PROGRAM 1-1

2. USER'S GUIDE 2-1

    2.1 PROGRAM MODES 2-1

    2.2 PROGRAM OUTPUT. 2-1

    2.3 LIMITATIONS PER PROBLEM 2-1

    2.4 ORDER OF CARDS IN JOB DECK. 2-2

    2.5 CARD PREPARATION 2-2

Appendix

A. PROGRAM LISTING. A-3

B. SAMPLE INPUT DATA. B-3

C. SAMPLE OUTPUT DATA C-3

1. GENERAL DESCRIPTION OF THE COMPUTER PROGRAM

This computer program is used to implement the Cate-Liebig algorithm for continuous data. The Cate-Liebig approach is intended to provide an alternative to stepwise multiple linear regression for the development of predictive models. In the program, cases are allocated to individual variables in an iterative manner based on the zero means of optimum binary (0-1) dummy variables created by using the Cate-Nelson critical level algorithm. The means of the allocated cases, along with a calculated overall maximum applied at the final critical level, are utilized to define a linear response surface with an independent equation for each variable.

The predicted value for any case is the minimum of all the values predicted for that case by the various individual equations. Variables are deleted from the model if less than a user-specified number of cases is allocated to that variable. Provision is made for testing the model on an independent set of data.

This documentation and the program structure are modeled after the widely known biomedical (BMD) computer programs developed by the University of California. It is intended to be readily usable for persons familiar with the BMD program package; however, this documentation may be used without additional reference to any other publication.

2. USER'S GUIDE

2.1 PROGRAM MODES

The program may be run in two primary modes. In the first mode, the program fits a set of dependent data using the Cate-Liebig algorithm and optionally provides a test of the resulting model on an independent data set. In the second mode, the program runs a user-specified model on an independent data set.

2.2 PROGRAM OUTPUT

Output from this program includes the following:

1. Summary of allocations.
2. Model coefficients.
3. For fit and test data (if any):
   a. Limiting variable for each case.
   b. Estimate, actual value, and error of dependent variable.
   c. Simple error statistics, including mean error, mean square error, root mean square error, and coefficient of determination.
   d. Plot of estimated versus actual value of the dependent variable.
4. Optional output, such as
   a. Plot of raw data for each variable.
   b. Summary of consistency check.
   c. Summary of critical-level splits after each iteration.
   d. Plot of cases allocated to each variable.

2.3 LIMITATIONS PER PROBLEM

Limitations per problem are as follows:

1. Number of original variables (NOV); $1 \leq NOV \leq 80$

2. Total number of variables after transgeneration (IP); $1 \leq IP \leq 80$

3. Number of variable format cards (NVFC); $1 \leq NVFC \leq 10$

4. Number of cases in fit (N); $2 \leq N \leq 2000$

5. Number of test cases (NTC); $1 \leq NTC \leq 2000$

6. Number of transgeneration cards (NTGC); $0 \leq NTGC \leq 99$

This program allows transgeneration of variables.

2.4 ORDER OF CARDS IN JOB DECK

Cards in the job deck should appear in the following order (NOTE: items b through g may be repeated as often as desired):

a. Program deck (a listing of the complete Fortran program is given in appendix A. All of the following cards are data read by that program.)

b. Problem card

c. Transgeneration cards (optional)

d. "Labels" cards (optional)

e. F-type variable format card(s)

f. Data deck (place data input deck here, including test cases)

g. Control-delete cards (optional). Alternatively when run in the model test mode only, the model definition cards are input at this point.

h. Finish card

2.5 CARD PREPARATION

a. System cards (to be determined)

b. Problem card:

Column	Problem
1-6	Problem (mandatory)
10-15	Alphanumeric problem name
17-20	Number of cases

Column	Description
24-25	Number of original variables
29-30	Number of transgeneration cards
34-35	Number of variables added by transgeneration
39-40	Index of dependent variable
44-45	Number of control-delete cards
49-50	Number of labeled variables
52-53	Minimum R^2 on split for allocation
54-55	Do not plot raw data if NPS = 1
57-58	Consistency check will not be used if = 1
59-60	Print summary of splits for each split if = 1
64-65	Do not print and plot each allocated variable if = 1
66-70	Number of test cases
71-72	Number of variable format cards
73-75	Smallest number of observations allowed for a variable to be allocated
76-78	Index of weighting vector

NOTE: Any input variable may be used. All cases are assumed to be weighted equally if index is not given a value.

If the number of cases = 0, the program is run in the model test mode and the model definition cards are required.

c. Transgeneration card:

Column	Description
1-6	TRNGEN
7-9	Variable index K
10-11	Code from transgeneration list
12-14	Variable index I
15-20	Variable index J or constant C

The term *transgeneration* is used to include transformations of input variables and creation of new variables prior to computation. Notations to be used in the transgeneration list include the following.

i, j, k = variable indices (need not be different)

c = a constant

$a_1, a_2, a_3, \cdots$ = constants

n = the number of cases, or sample size

Code	Transgeneration	Restriction
01	$\sqrt{X_i} \rightarrow X_k$	$X_i \geq 0$
02	$\sqrt{X_i} + \sqrt{X_i + 1} \rightarrow X_k$	$X_i \geq 0$
03	$\log_{10} X_i \rightarrow X_k$	$X_i \geq 0$
04	$e^{X_i} \rightarrow X_k$	
05	$\text{arc sin } \sqrt{X_i} \rightarrow X_k$	$0 \leq X_i \leq 1$
06	$\text{arc sin } \sqrt{X_i/(n + 1)}$	
	$+ \text{ arc sin } \sqrt{(X_i + 1)/(n + 1)} \rightarrow X_k$	$0 \leq (X_i/n) \leq 1$
07	$1/X_i \rightarrow X_k$	$X_i \neq 0$
08	$X_i + c \rightarrow X_k$	None
09	$X_i c \rightarrow X_k$	None
10	$X_i{}^c \rightarrow X_k$	$X_i \geq 0$
11	$X_i + X_j \rightarrow X_k$	None
12	$X_i - X_j \rightarrow X_k$	None
13	$X_i X_j \rightarrow X_k$	None
14	$X_i/X_j \rightarrow X_k$	$X_j \neq 0$
15	If $X_i > c$, $1 \rightarrow X_k$; otherwise $0 \rightarrow X_k$	None

Code	Transgeneration	Restriction
16	If $X_i \geq X_j$, $1 \rightarrow X_k$; otherwise $0 \rightarrow X_k$	None
17	$\log_e X_i \rightarrow X_k$	$X_i > 0$
18	If $X_i \leq c$, $1 \rightarrow X_k$; otherwise $0 \rightarrow X_k$	None
20	$\sin X_i \rightarrow X_k$	None
21	$\cos X_i \rightarrow X_k$	None
22	$\arctan X_i \rightarrow X_k$	None
23	$X_i^{X_j} \rightarrow X_k$	$X_i > 0$
24	$c^{X_i} \rightarrow X_k$	$c > 0$

When a violation of a restriction in the right-hand column occurs during transgeneration, the program will print a diagnostic message.

d. Label cards:

Column	Description
1-6	Labels (mandatory)
7-10	Number of the variable (or category or index) to be named (this number must be right-justified)
11-16	Corresponding alphanumeric name
17-20	Number of another variable
21-26	Corresponding alphanumeric name
⋮	
67-70	Number of another variable
71-76	Corresponding alphanumeric name of that variable (up to 7 variables per card)

Label cards allow the user to substitute alphanumeric names for the usual numeric indices that appear on the printed output. There may be from one to seven pairs of variable numbers and labels on each "labels" card. If desired, only one pair may be specified on each card. However, the total number of labels appearing on all the "labels" cards must be equal the number of labels specified on the problem card. It is not necessary to label all the variables. Those labeled may be listed in any order.

e. F-type variable format cards: The format is keypunched beginning with a left parenthesis, followed by a sequence of specifications, and closed by a right parenthesis. The first two specifications below are followed by a comma, except when they precede a slash or a right-hand parenthesis. Blank columns within the format are ignored, and columns 73 to 80 are not used. (A complete description of formats can be found in any Fortran programming manuals.) The specifications for F-type variable format cards are as follows:

- "nFw.d" is the floating point (decimal) indicator; n is the number of fields of width w (includes a sign and decimal point if punched); and d is the number of digits to the right of the decimal point if the decimal is not punched ($0 \leq d \leq w$). If the decimal is punched, d is ignored. If n is not specified, it is assumed to be 1.
- "mX" (alphabetic X), in which X is the skip indicator and m is the number of columns to be skipped.
- "/" (slash) indicates, "go to the next data card." Depending on its location in the format statement, the "/" will either direct the program to go immediately to the next card (ignoring any further information on the current card) or skip one card altogether. For example, if a format begins with "/," the program will automatically skip the first card, read the second, skip the third card, etc. If a format ends with "/," the program will automatically read the first card, skip the second, read the third card, etc.
- "//" indicates, "go to the card after next." Two slashes "//" will direct the program to skip two cards. Any number of slashes may be used.

f. Control-delete cards

Column	Description
1-6	CØNDEL
7	Control value for first variable
8	Control value for second variable
⋮	
72	Control value for 66th variable

If there are more than 66 variables, continue on another card of the same form until all variables have been specified. The variable numbers above refer to variables after transgeneration.

NOTE: The following control values are permissible:

0 — Linear form will be assumed.

1 — Delete variable (or dependent variable).

2 — A 0-1 form will be assumed.

If no control-delete cards are included or if a field is left blank on the control-delete cards included in the deck, the value 0 will be assigned if the variable is not the dependent variable and the value 1 will be assigned if it is the dependent variable.

g. Model definition cards: If the number of cases on the problem card is set equal to zero, the control delete card(s) is (are) replaced by Card 1:

Column	Description
1-3	Number of variables in the model (integer)
4-14	Maximum Y response in model (floating point)

This card is followed by one card for each variable in the model in the following format.

Column	Description
1-3	Variable number after transgeneration (integer)

Column	Description
4-14	Intercept (linear model) or lower mean 0-1 model (floating point)
15-25	Slope (floating point)
26-36	Critical level (floating point)
37-40	Type of response function for this variable (±1, 0)

If the code for the type of response is set equal to zero, a linear model of the $A + BX_i$ will be assumed. If the slope, B, is positive,

$$Y_i + A + BX_i \qquad X_i \leq \text{critical level}$$

$$Y_i = \text{maximum} \qquad \text{otherwise}$$

If the slope is negative,

$$Y_i = A + BX_i \qquad X_i > \text{critical level}$$

$$Y_i = \text{maximum} \qquad \text{otherwise}$$

If the code for the type of response is set equal to ±1, a 0-1 mean model will be assumed. If the code equal +1,

$$Y_i = \text{lower mean} \qquad X_i \leq \text{critical level}$$

$$Y_i = \text{maximum} \qquad \text{otherwise}$$

If the code equals -1,

$$Y_i = \text{lower mean} \qquad X_i > \text{critical level}$$

$$Y_i = \text{maximum} \qquad \text{otherwise}$$

Note that the program output tabulates all of these values in the model summary. When a 0-1 form has been assumed in the model fit, the appropriate lower mean value will be listed as the intercept.

The form of the response functions may be mixed on a variable-by-variable basis.

g. Finish card

Column	Description
1-6	Finish

This card will notify the program that the entire job is finished.

APPENDIX A

PROGRAM LISTING

APPENDIX A

PROGRAM LISTING

A listing of the complete Fortran program is given on the following pages.

```
C
C
C.......................................................................
C     THIS PROGRAM IMPLEMENTS THE CATE LIEBIG CRITICAL LEVEL ALGORITHM
C
C     PROBLEM CARD FORMAT
C      COL      NAME
C      1-6      XMAN PROBLM
C     10-15     CODE ALPHANUMERIC PROBLEM NAME
C     17-20     N    NUMBER OF CASES
C     24-25     NOV  NUMBER OF ORIGINAL VARIABLES
C     29-30     NTGC NUMBER OF TRANSGENERATION CARDS
C     34-35     NVA  NUMBER OF VARIABLES ADDED BY TRANSGENERATION
C     39-40     INDEXINDEX OF DEPENDENT VARIABLE
C     44-45     NCD  NUMBER OF CONTROL DELETE CARDS
C     49-50     NLV  NUMBER OF LABELED VARIABLES
C     52-53     R2   VARIABLES WITH RSQ LESS THAN R2 WILL NOT
C                    BE ALLOCATED
C     54-55     NPS  DO NOT PLOT RAW DATA IF NPS = 1
C     57-58     NCON CONSISTENCY CHECK WILL NOT BE USED IF NCON = 1
C     59-60     NS   PRINT SUMMARY OF SPLITS FOR EACH SPLIT IF NS = 1
C     64-65     NPA  DO NOT PRINT AND PLOT EACH ALLOCATED VARIABLE IF NPA
C                    = 1
C     66-70     NTC  NUMBER OF TEST CASES
C     71-72     NVFC NUMBER OF VARIABLE FORMAT CARDS
C     73-75     NAL  SMALLEST NUMBER OF OBSERVATIONS ALLOWED FOR A VARIABL
C                    BE ALLOCATED
C     76-78     IDWTGINDEX OF WEIGHTING VECTOR
C.......................................................................
C
C
      DIMENSION KTRANS(3,100),TRANS(100),RES(200),Y(080),XX(2000,3)
     1     ,C(160),ALBEL(160),ALI(160,9),CC(160),AL(160,10)
      DIMENSION XMI(2),XMA(2),YPLOT(4000),M(2000) ,X(0500,80)
      COMMON NTGC,TRANS,KTRANS,Y,M,NINCS,ALI,AL,C,  XMI,XMA,YPLOT
      INTEGER C,ALBEL,CC,D
      DATA PROBLM/4HPROB/
      DATA TRNGEN/4HTRNG/
      DATA CONDEL/4HCOND/
      DATA FINISH/4HFINI/
      NST = 0
C
C     READ PROGRAM CARD
C
1      READ(5,100) XMAN,CODE,N,NOV,NTGC,NVA,INDEX,NCD,NLV,R2,NPS,NCON
     1,NS,NPA,NTC,NVFC,NAL,IDWTG
100   FORMAT(A4,5X,2X,A4,1X,I4,6(3X,I2),1X,F2.2,I2,1X,2I2,3X,I2,I5,I2,2I
     13)
      IF(XMAN-FINISH)2,9001,2
2     IF(XMAN-PROBLM)9002,3,9002
9001  WRITE(6,8020)
8020  FORMAT(///24H FINISH CARD ENCOUNTERED)
9999  WRITE(6,8021)
8021  FORMAT(19H PROGRAM TERMINATED)
      CALL EXIT
9002  WRITE(6,8022)
8022  FORMAT(43H NEITHER PROBLM NOR FINISH CARD ENCOUNTERED)
      GO TO 9999
9004  WRITE(6,8024)
8024  FORMAT(37H CARD INCORRECTLY PUNCHED OR MISSING.)
      GO TO 9999
3     IP=NOV+NVA
      WRITE(6,8003) CODE,N,NOV, INDEX,NVFC
8003  FORMAT(37H1 CATE LIEBIG CRITICAL LEVEL ANALYSIS//2X,12HPROGRAM COD
     1E,17X,A6/2X,15HNUMBER OF CASES,16X,I4/2X,28HNUMBER OF ORIGINAL VAR
     2IABLES,5X,I2           /2X,27HINDEX OF DEPENDENT VARIABLE,6X,I2/2X,
     5                     31HNUMBER OF VARIABLE FORMAT CARDS,2X,I2//)
      IF(NTC.NE.0) WRITE(6,7013)  NTC
7013  FORMAT(2X,21HNUMBER OF TEST CASES ,I5)
      IF(NOV.NE.0) WRITE(6,7011) NVA,IP
7011  FORMAT(2X,45HNUMBER OF VARIABLES ADDED BY TRANSGENERATION ,I2,2X,
     126HTOTAL NUMBER OF VARIABLES ,I3)
      IF(NLV.NE.0) WRITE(6,7012)
7012  FORMAT(2X,21HVARIABLES ARE LABELED)
      IF(NCD.NE.0) WRITE(6,7005)
7005  FORMAT(2X,25HCONTROL DELETE CARDS USED )
      IF(NPS.NE.1) WRITE(6,7006)
7006  FORMAT(2X,13HPLOT RAW DATA   )
```

```
      IF(NPA.NE.1) WRITE(6,7008)
7008  FORMAT(2X,30HPRINT AND PLOT EACH ALLOCATION )
      IF(NS.EQ.1) WRITE(6,7004)
7004  FORMAT(2X,41HPRINT SUMMARY OF SPLIT FOR EACH ITERATION)
      IF(NAL.NE.0) WRITE(6,7009) NAL
7009  FORMAT(2X,47HALLOCATIONS LIMITED TO VARIABLES WITH AT LEAST ,I4,2X
     1,12HOBSERVATIONS )
      IF(R2.GE.0.001) WRITE(6,9011) R2
9011  FORMAT(2X,57HALLOCATIONS LIMITED TO VARIABLES WITH A RSQ GREATER T
     1HAN ,F4.2)
      IF(NCON.EQ.1) WRITE(6,9012)
9012  FORMAT(2X,26HCONSISTENCY CHECK NOT USED )
      IF(IDWTG.NE.0) WRITE(6,7010) IDWTG
7010  FORMAT(2X,9HVARIABLE ,I4,2X,26HUSED AS A WEIGHTING FACTOR )
      IF(IT.EQ.1) WRITE(6,9003)
9003  FORMAT(2X,23HDATA INPUT IS FROM TAPE)
      IF(NOV.LE.80.AND.IP.LE.80) GO TO 7
      WRITE(6,8023)
8023  FORMAT(39H0NUMBER OF VARIABLES OUTSIDE OF LIMITS. )
      GO TO 9999
7     IF(NTGC.EQ.0) GO TO 108
      WRITE(6,7014)
7014  FORMAT(//2X,21HTRANSGENERATIONS USED //2X,18H K CODE  I   J OR C )
      DO 1000 I=1,NTGC
C
C     READ TRANSGENERATION CARD
C
      READ(5,8002)YMAN,(KTRANS(J,I),J=1,3),TRANS(I)
8002  FORMAT(A4,2X,I3,I2,I3,F6.0)
      IF(YMAN-TRNGEN) 9006,7777,9006
9006  WRITE(6,8026)
8026  FORMAT(16H0TRANSGENERATION)
      GO TO 9004
7777  KIP = IP
      IF(NVA.LT.0) KIP = NOV
      IF(KTRANS(1,I).GT.KIP.OR.KTRANS(3,I).GT.KIP) GO TO 9777
      IF(KTRANS(2,I).GT.24) GO TO 9775
      GO TO 1000
9777  WRITE(6,8777) I
8777  FORMAT(///51H VARIABLE NUMBER SPECIFIED ON TRANSGENERATION CARD ,I
     14,12H NOT CORRECT )
      GO TO 9999
9775  WRITE(6,8775) KTRANS(2,I)
8775  FORMAT(///29H ILLEGAL TRANSGENERATION CODE ,I3,10H SPECIFIED)
      GO TO 9999
1000  WRITE(6,7015) (KTRANS(J,I),J = 1,3),TRANS(I)
7015  FORMAT(3(2X,I2),2X,F7.2)
C
C     READ LABELS CARD
C
108   CALL RDLBL2(NLV,IP,ALBEL)
C
C     READ VARIABLE FORMAT CARDS
C
      IF(NVFC.LE.10) GO TO 106
      WRITE(6,4000)
4000  FORMAT(1H023X71HNUMBER OF VARIABLE FORMAT CARDS INCORRECTLY SPECIF
     XIED, ASSUMED TO BE 1.)
      NVFC = 1
106   NVFC=NVFC*20
      READ(5,8004) (RES(I),I=1,NVFC)
8004  FORMAT(20A4)
      REWIND 2
      N3 = N + NTC
      DO 112 K = 1,N3
      M(K) = 0
      NINCS=K
C
C     READ DATA
C
      READ(5,RES) (Y(L),L=1,NOV)
      IF(NTGC)11,11,12
C
C     TRANSGENERATION OF INPUT DATA
C
12    CALL TRANGN
11    CONTINUE
      WRITE(2,900) Y
```

```
1129  DO 1121 JJ = 1,IP
1121  X(K,JJ) = Y(JJ)
112   CONTINUE
      REWIND 2
      N1 = IP
      IF(N.EQ.0) GO TO 213
113   DO 360 I=1,160
      CC(I) = 0
360   C(I)=0
      NVAL = 0
      IF(NCD.EQ.0) GO TO 36
      IF(IP-66) 351,351,352
C
C     READ CONTROL/DELETE CARDS
C
351   READ(5,8011) VMAN,(C(I),I=1,IP)
8011  FORMAT(A4,2X,66I1)
      GO TO 361
352   READ(5,8011) VMAN,(C(I),I=1,66),DUM,(C(I),I=67,IP)
361   IF(VMAN-CONDEL) 9010,36,9010
9010  WRITE(6,8030)
8030  FORMAT(15HOCONTROL-DELETE)
      GO TO 9004
36    C(INDEX)=1
      DO 162 I = 1,IP
162   C(I+IP) = 1
      IF(NAL.LT.1) NAL = 1
      IF(IDWTG.NE.0) C(IDWTG) = 1
      IQ = IP*2
37    DO 371 I = 1,IQ
      DO 371 J = 1,9
371   ALI(I,J) = 0.0
C
C     BEGIN SPLIT
C
      DO 14  J=1,IP
      IF( C(J).EQ.1) GO TO 14
      D = 0
      DO 122 I=1,N
      DO 1221 K = 1,IP
1221  Y(K) = X(I,K)
      IF(M(I).NE.0) GO TO 122
      D = D + 1
      XX(D,3) = 1.0
      IF(IDWTG.NE.0) XX(D,3) = Y(IDWTG)
      YPLOT(D) = Y(INDEX)
      YPLOT(D+N) = Y(J)
      XX(D,1)=Y(J)
      XX(D,2)=Y(INDEX)
122   CONTINUE
      N2 = D
      IF(D.LT.NAL) GO TO 21
      IF(D.EQ.1) GO TO 21
      N3=N2 - 1
      IF(NPS.EQ.1) GO TO 121
C
C     PLOT RAW DATA
C
      WRITE(6,1211) J,ALBEL(J)
1211  FORMAT(1H1,22HRAW DATA FOR VARIABLE ,I4,2X,A6/)
      CALL SETPLT(N)
121   K=0
C
C     SORT DATA INTO ASCENDING ORDER
C
      DO 13  I=1,N3
      A =XX(I,1)
      AA=XX(I,2)
      AAA = XX(I,3)
      B =XX(I+1,1)
      BB=XX(I+1,2)
      BBB = XX(I+1,3)
      IF(B.GE.A) GO TO 13
      K=1
      XX(I,1) = B
      XX(I,2)=BB
      XX(I,3) = BBB
      XX(I+1,1)=A
```

```
         XX(I+1,2)=AA
         XX(I+1,3) = AAA
13       CONTINUE
         IF(K.EQ.1) GO TO 121
         DO 133 II= 1,N3
         NN=II
         XA = 0.0
         XB = 0.0
         XBAR    =0.0
         YBAR    =0.0
         SUMXY   =0.0
         SUMXSQ  =0.0
         SUMYSQ  =0.0
         Y1=0.0
         Y2=0.0
         Y1C = 0.0
         Y2C = 0.0
         NNN= NN+1
         XN=0
         IF(XX(NN,1).GE.XX(NNN,1)) GO TO 133
         DO 135 I= 1,NN
         XN=XN+XX(I,3)
         XA = XA + XX(I,1)*XX(I,3)
         Y1=Y1+XX(I,2)*XX(I,3)
         Y1C = Y1C + XX(I,3)
135      YBAR=YBAR+XX(I,2)*XX(I,3)
         DO 136 I=NNN,N2
         XN=XN+XX(I,3)
         XBAR=XBAR+XX(I,3)
         XB = XB + XX(I,1)*XX(I,3)
         Y2 = Y2+XX(I,2)*XX(I,3)
         Y2C = Y2C + XX(I,3)
136      YBAR=YBAR+XX(I,2)*XX(I,3)
         XBAR=XBAR/XN
         YBAR=YBAR/XN
         Y1=Y1/Y1C
         Y2=Y2/Y2C
         XA = XA/Y1C
         XB = XB/Y2C
         DO 137 I= 1,NN
         SUMXY=SUMXY + (0.0-XBAR)*(XX(I,2)-YBAR)*XX(I,3)
         SUMXSQ=SUMXSQ +((0.0-XBAR)**2)*XX(I,3)
137      SUMYSQ=SUMYSQ +((XX(I,2)-YBAR)**2)*XX(I,3)
         DO 138 I=NNN,N2
         SUMXY=SUMXY + (1.0-XBAR)*(XX(I,2)-YBAR)*XX(I,3)
         SUMXSQ=SUMXSQ +((1.0-XBAR)**2)*XX(I,3)
138      SUMYSQ=SUMYSQ +((XX(I,2)-YBAR)**2)*XX(I,3)
         R = SQRT(SUMXSQ*SUMYSQ)
         IF(R.GT.0.001) R = SUMXY/R
         RSQ=R**2
         IF(RSQ.LT.ALI(J     ,7)) GO TO 133
         ALI(J     ,7)=RSQ
         ALI(J     ,2)=XX(II,1)
         ALI(J     ,1)=1.0
7234     ALI(J,9) = (XX(II+1,1) + XX(II,1))/2.0
         ALI(J,8) = XA
         I1=3
         I2=5
         IF(Y2.GT.Y1) GO TO 1382
         ALI(J     ,1)=-1.0
7235     ALI(J,9) = (XX(II+1,1) + XX(II,1))/2.0
         ALI(J,8) = XB
         I1=5
         I2=3
1382     ALI(J     ,I1)=Y1
         ALI(J     ,I2)=Y2
         ALI(J     ,I1+1)=NNN-1
         ALI(J     ,I2+1)=N2-NN
133      CONTINUE
14       CONTINUE
         IF(NS.NE.1) GO TO 168
         WRITE(6,103)
103      FORMAT(1H1,4X,17HSUMMARY OF SPLITS//)
         WRITE(6,107)
107      FORMAT(//23X,8HCRITICAL,5X,6H0 MEAN,5X,6HNUMBER,9X,6HNUMBER/9X,
        163HVARIABLE SIGN  LEVEL       X         Y     CASES  1 MEAN  CASES RSQ
        2)
         DO 15 I=1,IP
```

```
      IF( C(I).EQ.1) GO TO 15
      WRITE(6,104)  ALBEL(I),I,(ALI(I,J),J=1,2),ALI(I,8),(ALI(I,J),J=3,7)
104   FORMAT(2X,A6,3X,I3,F8.0,F8.2,F9.2,F8.2,F5.0,F9.2,F6.0,F5.2,10F2.0)
15    CONTINUE
C
C     BEGIN ALLOCATION
C
168   XMIN=99998.0
      IVAR=-1
      N1=IP
      DO 17  I= 1,IQ
      IF( C(I).EQ.1)  GO TO 17
      IF(ALI(I,7).LT.R2)  GO TO 17
      IF(CC(I).NE.0)  GO TO 170
169   IF(ALI(I,4).LT.NAL)  GO TO 17
      GO TO 171
170   IVA = ALI(I,1)
      IF(IVA.EQ.CC(I))  GO TO 169
      IF(ALI(I,4).LT.NAL)  GO TO 17
      IF(ALI(I,3).GT.XMIN)  GO TO 17
      IVAR = I + IP
      GO TO 176
171   IF(ALI(I,3).GT.XMIN)  GO TO 17
      IVAR=I
176   IV = I
      XMIN=ALI(I,3)
17    CONTINUE
      IF(IVAR.EQ.-1)  GO TO 21
      C(IVAR) = C(IV)
      CC(IVAR)=ALI(IV,1)
      KVAR=IVAR
      IF(IVAR.GT.N1)  KVAR = IVAR - N1
      DO 191 I=1,N
      DO 1791 K = 1,IP
1791  Y(K) = X(I,K)
      IF(M(I).NE.0)  GO TO 191
      IF(ALI(KVAR,1).LT.0.0)  GO TO 183
C
C  VARIABLE ALLOCATED POSITIVE
C
      IF(Y(KVAR)    .GT.ALI(KVAR,2))  GO TO 191
179   M(I) = IVAR
      GO TO 191
C
C  VARIABLE ALLOCATED NEGATIVE
C
183   IF(Y(KVAR)    .LE.ALI(KVAR,2))  GO TO 191
      GO TO 179
191   CONTINUE
      NVAL=NVAL+1
      AL(NVAL,1)=IVAR
      DO 20 I=1,9
20    AL(NVAL,I+1)=ALI(KVAR,I)
      NPS = 1
      GO TO 37
C
C     INITIAL ALLOCATION COMPLETE
C
21    WRITE(6,109)
109   FORMAT(1H1,4X,19HINITIAL ALLOCATIONS)
      WRITE(6,107)
      DO 22 I=1,NVAL
      L = AL(I,1)
      J = L
      IF(L.GT.N1) L = L - N1
22    WRITE(6,104)  ALBEL(L),J,(AL(I,JJ),JJ=2,3),AL(I,9),(AL(I,JJ),JJ=4,8
     1)
      WRITE(6,1092) NAL  ,R2
1092  FORMAT(///5X,25HVARIABLES WITH LESS THAN ,I3,26H CASES WERE NOT AL
     1LOCATED./5X,45HVARIABLES WITH SPLITS HAVING A RSQ LESS THAN ,F4.2,
     120H WERE NOT ALLOCATED.)
      DO 221 I = 1,IQ
      DO 221 J = 1,9
221   ALI(I,J) = 0.0
      DO 222 I = 1,NVAL
      L = AL(I,1)
      ALI(L,9) = AL(I,10)
      ALI(L,5) = AL(NVAL,6)
```

```
      ALI(L,1) = AL(I,2)
      ALI(L,2) = AL(I,3)
      ALI(L,3) = (ALI(L,3)*ALI(L,4) + AL(I,4)*AL(I,5))/(ALI(L,4) + AL(I,
     15))
      ALI(L,8) = (ALI(L,8)*ALI(L,4) + AL(I,9)*AL(I,5))/(ALI(L,4) + AL(I,
     15))
222   ALI(L,4) = ALI(L,4) + AL(I,5)
C
C     CONSISTENCY CHECK
C
      DO 2231 I = 1,IQ
      L = ALI(I,1)
2231  IF(L.EQ.0) C(I) = 1
      IF(NCON.EQ.1) GO TO 2126
      DO 2220 L = 1,NVAL
      I = AL(L,1)
      IF(C(I).EQ.2) GO TO 2120
      ALI(I,7) =(ALI(I,5) - ALI(I,3))/(ALI(I,9) - ALI(I,8))
9120  ALI(I,6) = ALI(I,3) - ALI(I,7)*ALI(I,8)
      GO TO 2220
2120  ALI(I,7) = 0.0
      ALI(I,6) = ALI(I,3)
2220  CONTINUE
      WRITE(6,2219)
2219  FORMAT(1H1,4X,17HCONSISTENCY CHECK//54X,8HORIGINAL,20X,3HNEW/5X,
     186HCASE    ORIGINAL    NEW            ACTUAL Y        LIMITING X   EST Y
     2        LIMITING X     EST Y   )
      DO 2222 I = 1,N
      XM1 = AL(NVAL,6)
      JJ = 0
      DO 2232 K = 1,IP
2232  Y(K) = X(I,K)
      DO 2221 J = 1,IQ
      IF(C(J).EQ.1) GO TO 2221
      K = J
      IF(J.GT.N1) K = J - N1
      XM = ALI(J,6) + ALI(J,7)*Y(K)
      IF(ALI(J,1).EQ.-1.AND.Y(K).LE.ALI(J,2)) XM = AL(NVAL,6)
      IF(ALI(J,1).EQ.1.AND.Y(K).GT.ALI(J,2)) XM = AL(NVAL,6)
      IF(XM.GT.XM1) GO TO 2221
      XM1 = XM
      JJ = J
2221  CONTINUE
      IF(M(I).EQ.JJ) GO TO 2222
      J = M(I)
      IF(J.EQ.0) GO TO 2222
      K=J
      IF(J.GT.N1) K=J-N1
      XM2=ALI(J,6)+ALI(J,7)*Y(K)
      KK = JJ
      IF(JJ.GT.N1) KK = JJ - N1
      WRITE(6,2218) I,J,JJ,Y(INDEX),Y(J),XM2,Y(KK),XM1
2218  FORMAT(5X,I4,4X,I6,3X,I6,5X,F10.5,2(5X,2F10.5))
      AI79=ALI(J,4)-1.0
      IF(AI79.LE..1) GO TO 2318
      ALI(J,3) = (ALI(J,3)*ALI(J,4) - Y(INDEX))/(ALI(J,4) - 1.0)
      ALI(J,8) = (ALI(J,8)*ALI(J,4) - Y(K))/(ALI(J,4) - 1.0)
      ALI(J,4) = ALI(J,4) - 1.0
      GO TO 2319
 2318 ALI(J,3)=0.
      ALI(J,8)=0.
      ALI(J,4.)=0.
      C(J)=1
 2319 ALI(JJ,3) = (ALI(JJ,3)*ALI(JJ,4) + Y(INDEX))/(ALI(JJ,4) + 1.0)
      K = JJ
      IF(JJ.GT.N1) K = JJ - N1
      ALI(JJ,8) = (ALI(JJ,8)*ALI(JJ,4) + Y(K))/(ALI(JJ,4) + 1.0)
      ALI(JJ,4) = ALI(JJ,4) + 1.0
      M(I) = JJ
2222  CONTINUE
      WRITE(6,2225)
2225  FORMAT(1H1,4X,17HFINAL ALLOCATIONS)
      WRITE(6,107)
      DO 2226 I = 1,IQ
      J = I
      IF(I.GT.N1) J = I - N1
       IF(C(I).EQ.1) GO TO 2226
       WRITE(6,104) ALBEL(J),I,(ALI(I,JJ),JJ = 1,2),ALI(I,8),(ALI(I,JJ),
```

```
     1JJ=3,5)
2226  CONTINUE
2126  IF(NPA.EQ.1) GO TO 212
      DO 211 J = 1,IQ
      IF(C(J).EQ.1) GO TO 211
      JZ = J
      IF(J.GT.N1) JZ = J - N1
      WRITE(6,105) J,ALBEL(JZ),ALI(J,4)
105   FORMAT(1H1,4X,20HVARIABLE ALLOCATED  ,I3,2X,A6,2X,F5.0,2X,5HCASES/
     1)
      WRITE(6,1051)
1051  FORMAT(//5X,4HCASE,8X,1HX,12X,1HY)
      L = 0
      L1 = ALI(J,4)
      DO 210 I = 1,N
      DO 2101 K = 1,IP
2101  Y(K) = X(I,K)
       IF(M(I).NE.J) GO TO 210
      WRITE(6,1052) I,Y(JZ)   ,Y(INDEX)
1052  FORMAT(5X,I4,2F13.4,F20.2)
      L = L + 1
      YPLOT(L) = Y(INDEX)
      YPLOT(L+N) = Y(JZ)
210   CONTINUE
      WRITE(6,105) J,ALBEL(JZ),ALI(J,4)
      CALL SETPLT(L1)
211   CONTINUE
      GO TO 212
213   READ(5,214) NVAL,AL(NVAL,6)
214   FORMAT(I3,F11.5)
      IQ = 2*IP
      DO 215 I = 1,IQ
215   C(I) = 1
      DO 217 I = 1,NVAL
      READ(5,216) J,ALI(J,8),ALI(J,9),ALI(J,2),ALI(J,1)
216   FORMAT(I3,3F11.5,F4.0)
217   C(J) = 0
212   WRITE(6,110)
110   FORMAT(1H1,4X,13HMODEL SUMMARY/5X,55HVARIABLE NUM     INTERCEPT
     1     SLOPE    CRITICAL LEVEL   )
      DO 224 I = 1,IQ
      L = I
      IF(I.GT.N1) L=I-N1
      IF( C(I).EQ.1) GO TO 224
      IF(N.EQ.0) GO TO 2242
      IF(C(I).EQ.2) GO TO 2241
      ALI(I,9) = (ALI(I,5) - ALI(I,3))/(ALI(I,9) - ALI(I,8))
      ALI(I,8) = ALI(I,3) - ALI(I,9)*ALI(I,8)
      GO TO 2242
2241  ALI(I,9) = 0.0
      ALI(I,8) = ALI(I,3)
2242  CONTINUE
      WRITE(6,111) ALBEL(L),I,ALI(I,8),ALI(I,9),ALI(I,2)
  224 CONTINUE
111   FORMAT(6X,A6,2X,I3,2(2X,F11.5),6X,F11.5)
      WRITE(6,114) AL(NVAL,6)
114   FORMAT(//5X,7HMAXIMUM,F11.5)
      IF(N.EQ.0) GO TO 218
      WRITE(6,1091)
1091  FORMAT(1H1,4X,14HDEPENDENT DATA  //)
      N5 = 1
      CALL TEST(N,INDEX,NVAL,N1,N5)
218   IF(NTC.EQ.0) GO TO 1
      WRITE(6,1090)
1090  FORMAT(1H1,4X,9HTEST-DATA//)
      N5 = N + 1
      CALL TEST(NTC,INDEX,NVAL,N1,N5)
      GO TO 1
  900 FORMAT(5E15.6)
      END
      SUBROUTINE TEST(NTC,INDEX,NVAL,N1,N5)
      DIMENSION ALI(160,9),AL(160,10),TRANS(100),KTRANS(3,100),Y(080)
      DIMENSION XMI(2),XMA(2),YPLOT(4000),C(160)
      INTEGER C
      COMMON NTGC,TRANS,KTRANS,Y,N,NINCS,ALI,AL,C,   XMI,XMA,YPLOT
      WRITE(6,101)
101   FORMAT(5X,60HCASE  VARIABLE   LIMITING X      EST Y       ACT Y
     1  ERROR)
```

FILE. . .   CL0516   FORTRAN P1

```
      YBAR = 0.0
      REWIND 2
      IF(N5.EQ.1) GO TO 221
      N6=N5-1
      DO 220 I=1,N6
220   READ(2,900)Y
221   CONTINUE
      DO 22 I = 1,NTC
      READ(2,900) Y
22    YBAR = YBAR + Y(INDEX)
      YBAR = YBAR/NTC
      TSS = 0.0
      XBAR = 0.0
      RMSE = 0.0
      REWIND 2
      IF(N5.EQ.1)  GO TO 231
      DO 230 I = 1,N6
230   READ(2,900) Y
231   IF(N5.EQ.1) N = NTC
23    DO 241 I=1,NTC
      READ(2,900) Y
      XMIN=AL(NVAL,6)
      JJ=INDEX
      IQ = 2*N1
      DO 24 J = 1,IQ
      IF(C(J).EQ.1) GO TO 24
      K = J
      IF(J.GT.N1) K = J - N1
      XM = ALI(J,8) + ALI(J,9)*Y(K)
      IF(ALI(J,1).EQ.1.0.AND.Y(K).GT.ALI(J,2)) XM = AL(NVAL,6)
      IF(ALI(J,1).EQ.-1.0.AND.Y(K).LE.ALI(J,2)) XM = AL(NVAL,6)
      IF(XM.GT.XMIN) GO TO 24
      XMIN = XM
      JJ=K
24    CONTINUE
      TSS = TSS + (YBAR - Y(INDEX))**2
      ERROR=Y(INDEX)   -XMIN
      XBAR = XBAR + ERROR
      RMSE = RMSE + ERROR*ERROR
      Z=Y(JJ)
      YPLOT(I) = Y(INDEX)
      YPLOT(I+N) = XMIN
      IF(JJ.EQ.INDEX) Z        =99999.0
      WRITE(6,1091) I,JJ,Z        , XMIN,Y(INDEX)   ,ERROR
1091  FORMAT(6X,I3,5X,I3,3(3X,F10.4),F10.4)
241   CONTINUE
      XBAR = XBAR/NTC
      ESS = TSS - RMSE
      RSQ = ESS/TSS
      XMSE = RMSE/NTC
      RMSE = SQRT(XMSE)
      WRITE(6,1092) XBAR,RMSE,XMSE,RSQ
1092  FORMAT(//5X,10HMEAN ERROR,14X,F6.2/5X,22HROOT MEAN SQUARE ERROR,2X
     1,F6.2/5X,17HMEAN SQUARE ERROR,7X,F6.2/
     2                                   5X,3HRSQ,21X,F6.2)
      WRITE(6,102)
102   FORMAT(1H1,5X,39HACTUAL Y ON Y-AXIS, ESTIMATE ON X-AXIS.  )
      CALL SETPLT(NTC)
      RETURN
  900 FORMAT(5E15.6)
      END
      SUBROUTINE RDLBL2(NLBVAR,NVAR,ARRAY)
            INTEGER ALABEL,ARRAY,ITEST,DUMY,BLANK
      EQUIVALENCE(LABEL,ALABEL)
      DIMENSION ARRAY(80),IDUM(7),DUMY(7)
      DATA ALABEL/3HLAB/
      DATA BLANK/4H    /
      DO 1 I=1,NVAR
1     ARRAY(I)=BLANK
      IF(NLBVAR) 9,9,2
2     N=0
20    READ(5,3)ITEST,(IDUM(J),DUMY(J),J=1,7)
3     FORMAT(A3,3X,7(I4,A6))
      IF  (ITEST-LABEL) 4,6,4
4     WRITE(6,5)
5     FORMAT(36H0LABELS CARD NOT FOUND WHEN EXPECTED)
      CALL EXIT
6     DO 8 J=1,7
```

FILE. . . CL0516 FORTRAN P1

```
      K=IDUM(J)
      IF(K) 11,8,10
10    IF(K-NVAR) 7,7,11
11    WRITE(6,12)K,DUMY(J)
12    FORMAT(18HOLABELS CARD INDEX,I7,18H INCORRECT. LABEL ,A6,9H IGNORE
     1D.)
      GO TO 13
7     ARRAY(K)=DUMY(J)
13    N=N+1
      IF(N-NLBVAR) 8,9,9
8     CONTINUE
      GO TO 20
9     RETURN
      END
      SUBROUTINE TRANGN
      DIMENSION        TRANS(100),KTRANS(3,100),ALI(160,9),AL(160,10)
      DIMENSION X(080),              C(160),XMI(2),XMA(2),YPLOT(4000)
      COMMON NTGC,TRANS,KTRANS,X,K,NINCS,ALI,AL,C,  XMI,XMA,YPLOT
      INTEGER C
      ASNF(XX)=ATAN (XX/SQRT (1.0-XX**2))
      NCASE = K
      DO 100 I=1,NTGC
      M=KTRANS(1,I)
      N=KTRANS(3,I)
      NTRANS=KTRANS(2,I)
      IF(M-81)  91,91,99
91    IF(N-81)  92,92,99
92    IF((NTPANS-25)*NTRANS)50,99,99
99    WRITE( 6,199)I
199   FORMAT(22H TRANSGENERATION CARD ,I3,27H MISPUNCHED OR OUT OF ORDER
     1)
      GO TO 100
50    GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23
     1,24),NTRANS
1     IF(X(N)) 198,107,108
107   X(M)=0.0
      GO TO 100
108   X(M)= SQRT  (X(N))
      GO TO 100
2     IF(X(N)) 198,111,112
111   X(M)=1.0
      GO TO 100
112   X(M)= SQRT  (X(N))+SQRT  (X(N)+1.0)
      GO TO 100
3     IF(X(N)) 198,198,114
114    X(M)  = ALOG(X(N))*0.4342944819
      GO TO 100
4     X(M)=EXP  (X(N))
      GO TO 100
5     IF(X(N)) 198,107,117
117   IF(X(N)-1.0) 118,119,119
118   E=SQRT  (X(N))
      X(M)=ASNF(E)
      GO TO 100
119    X(M)  = 3.1415926536/2.0
      GO TO 100
6     FN=NCASE
      E=X(N)/(FN+1.0)
      B=F+1.0/(FN+1.0)
      IF(E) 198,123,124
123   IF(B) 198,107,127
127   X(M)=ASNF(SQRT  (B))
      GO TO 100
124   IF(B) 198,128,129
128   X(M)=ASNF(SQRT  (E))
      GO TO 100
129   E=SQRT  (E)
      B=SQRT  (B)
      X(M)=ASNF(E)+ASNF(B)
      GO TO 100
7     IF(X(N)) 131,198,131
131   X(M)= 1.0/X(N)
      GO TO 100
8     X(M)=X(N)+ TRANS( I )
      GO TO 100
9     X(M)=X(N)* TRANS( I )
      GO TO 100
10    IF(X(N)) 198,107,133
```

FILE. . .   CL0516   FORTRAN P1

```
133    X(M)= X(N)**TRANS( I)
       GO TO 100
11     NEWB= TRANS( I)
       X(M)= X(N)+X(NEWB)
       GO TO 100
12     NEWB= TRANS( I)
       X(M)=X(N)-X(NEWB)
       GO TO 100
13     NEWB= TRANS( I)
       X(M)=X(N)*X(NEWB)
       GO TO 100
14     NEWB= TRANS( I)
       IF(X(NEWB)) 134,197,134
134    X(M)= X(N)/X(NEWB)
       GO TO 100
15     IF(X(N).GT.TRANS(I)) GO TO 111
       GO TO 107
16     XNEWB=TRANS( I)
       IF(X(N)-(XNEWB)) 107,111,111
17     IF(X(N))198,198,163
163    X(M)= ALOG(X(N))
       GO TO 100
18      IF(X(N).LE.TRANS(I)) GO TO 111
       GO TO 107
19     X(M) = 6.11*EXP((-1.762042621E05+5.597607915E03*X(N)-2.050772636*X
      1(N)**2)/(1.254162E05+273.0*X(N)))
       GO TO 100
20     X(M)= SIN (X(N))
       GO TO 100
21     X(M)= COS (X(N))
       GO TO 100
22     IF(X(N)-1.57079632) 186,186,198
186    IF(X(N)+1.57079632) 198,187,187
187    X(M)=ATAN (X(N))
       GO TO 100
23     NEWB= TRANS( I)
       IF(X(N)) 198,198,188
188    X(M)=X(N)**X(NEWB)
       GO TO 100
24     IF(TRANS(I)) 198,198,189
189    X(M)= TRANS(I)**X(N)
       GO TO 100
197    N=NEWB
198    WRITE(          6,201)N,NINCS,KTRANS(2,I),M
201    FORMAT(23H THE VALUE OF VARIABLE ,I4, 9H IN CASE ,I5,55H VIOLATED
      1THE RESTRICTIONS FOR TRANSGENERATION OF TYPE ,I3,1H./40H THE PROGR
      2AM CONTINUED LEAVING VARIABLE ,I4,11H UNCHANGED.)
100    CONTINUE
       RETURN
       END
       SUBROUTINE SETPLT(N)
       DIMENSION KTRANS(3,100),TRANS(100),W(080),ALI(160,9),C(160)
       DIMENSION XMI(2),XMA(2),YPLOT(4000),AL(160,10)
       COMMON NTGC,TRANS,KTRANS,W,M,NINCS,ALI,AL,C,  XMI,XMA,YPLOT
       INTEGER C
       NPV = N
       NC = 1
       K=1
       DO 330 I=1,2
       XMI(I)=99999999.0
       XMA(I)=-99999999.0
       DO 320 J=K,NPV
       XMI(I)= AMIN1(YPLOT(J),XMI(I))
  320  XMA(I)= AMAX1(YPLOT(J),XMA(I))
       K=K+M
  330  NPV=NPV+M
       YMAX=XMA(1)
       YMIN=XMI(1)
       XMAX=XMA(2)
       XMIN=XMI(2)
       DO 550 I=1,N
       Z = YPLOT(M+I)

520    Y = YPLOT(I)
  530  CALL PLOTR(Y,YMIN,YMAX,Z,    XMIN,XMAX,NC)
  550  CONTINUE
  570  NC=0
  580  CALL PLOTR(Y,YMIN,YMAX,Z,    XMIN,XMAX,NC)
```

```
      RETURN                                                            CAT07810
      END                                                               CAT07820
      SUBROUTINE PLOTR(X,ZMIN,ZMAX,Y,     WMIN,WMAX,NC)                 CAT07830
      DIMENSION Y(15),CLAB(12),GF(10),FMT(12),XY(51,101),SYM(15)        CAT07840
      INTEGER XY, BLANKS                                                CAT07850
      DATA SYM(1)/1H./                                                  CAT07860
      NP = 1                                                            CAT07870
      DATA TC,TP,BLANKS/1H.,1H+,1H /                                    CAT07880
      DATA GF/           4H 1X,,4H 2X,,4H 3X,,4H 4X,,4H 5X,,4H 6X,,     CAT07890
     14H 7X,,4H 8X,,4H 9X,,4H 10X/                                      CAT07900
      DATA FMT/'(17X',' ','5(F1','2.3,','8X)/','7X, ',' ','4(F1','2.3,','CAT07910
     1'8X),',','F12.','3)   '/                                          CAT07920
 8000  FORMAT(1H 6X5(F12.3,8X),F12.3/17X,5(F12.3,8X))                   CAT07930
 8010 FORMAT(1H F12.3,1X,103A1,F12.3)                                   CAT07940
 8020 FORMAT (1H 13X,103A1)                                             CAT07950
 8030 FORMAT(1H  14X,101A1)                                             CAT07960
 8040 FORMAT(15X,20(5H+....),1H+)                                       CAT07970
      DATA NCC/2/                                                       CAT07980
      IF(NCC) 10,200,10                                                 CAT07990
   10 KL=0                                                              CAT08000
      CALL SCALE(WMIN,WMAX,100.0,JY,YMIN,YMAX,YIJ)                      CAT08010
      YR=YMAX-YMIN                                                      CAT08020
   20 J=JY                                                              CAT08030
      IF(J*(J-10))70,30,30                                              CAT08040
   30 IF(KL)50,50,40                                                    CAT08050
   40 WRITE (6,8040)                                                    CAT08060
      IF(KL)170,170,50                                                  CAT08070
   50 CLAB(1)= YMIN                                                     CAT08080
      DO 60 I=2,11                                                      CAT08090
   60 CLAB(I)=CLAB(I-1)+YIJ                                             CAT08100
      WRITE (6,8000)(CLAB(I),I=1,11,2),(CLAB(J),J=2,10,2)               CAT08110
      IF(KL)40,40,430                                                   CAT08120
   70 IF(J-5)90,100,80                                                  CAT08130
   80 J=J-5                                                             CAT08140
   90 JYT=5-J                                                           CAT08150
  100 CONTINUE                                                          CAT08160
  110 FMT(2)=GF(JY)                                                     CAT08170
      IF (KL) 120,120,140                                               CAT08180
  120 FMT(7)=GF(JY)                                                     CAT08190
      TT=JY                                                             CAT08200
      TT=TT*YIJ/10.0                                                    CAT08210
      CLAB(1)= YMIN+TT                                                  CAT08220
      DO 130 I=2,10                                                     CAT08230
  130 CLAB(I)=CLAB(I-1) +YIJ                                            CAT08240
      WRITE (6,FMT)(CLAB(I),I=2,10,2),(CLAB(I),I=1,9 ,2)                CAT08250
      IF(KL)140,140,430                                                 CAT08260
  140 IF(JY-5)150,160,150                                               CAT08270
  150 WRITE(6,8030)(TC,I=1,J     ),(TP,(TC,I=1,4),K=1,19),TP,(TC,I=1,JYT)CAT08280
      IF (KL) 170,170,120                                               CAT08290
  160 WRITE (6,8040)                                                    CAT08300
      IF (KL) 170,170,120                                               CAT08310
  170 CONTINUE                                                          CAT08320
      NCC=0                                                             CAT08330
      IC=0                                                              CAT08340
      IF(NP)280,180,180                                                 CAT08350
  180 DO 190 I=1,51                                                     CAT08360
       DO 190 J=1,101                                                   CAT08370
  190 XY(I,J)=BLANKS                                                    CAT08380
      CALL SCALE (ZMIN,ZMAX,50.0,JX,XMIN,XMAX,XIJ)                      CAT08390
      XR=XMAX-XMIN                                                      CAT08400
      GO TO 200                                                         CAT08410
      ENTRY PLOTS                                                       CAT08420
  200 IF(NC)420,360,210                                                 CAT08430
  210 IF(NP)280,220,220                                                 CAT08440
  220 DO 270 N=1,NC                                                     CAT08450
      SYMB=SYM(N)                                                       CAT08460
      XDIFFR=XMAX-X                                                     CAT08470
      IF(XDIFFR)230,240,240                                             CAT08480
  230 XDIFFR=0.0                                                        CAT08490
  240 YDIFFR=YMAX-Y(N)                                                  CAT08500
      IF(YDIFFR)250,260,260                                             CAT08510
  250 YDIFFR=0.0                                                        CAT08520
  260 L=51.0-(50.0*XDIFFR)/XR+.5                                        CAT08530
      K=101.0-(100.0*YDIFFR)/YR+.5                                      CAT08540
      CALL FORM2(SYMB,XY(L,K))                                          CAT08550
  270  CONTINUE                                                         CAT08560
      GO TO 440                                                         CAT08570
  280 DO 290 I=1,101                                                    CAT08580
```

```
  290 XY(1,I)=BLANKS
      L=1
      DO 320 N=1,NC
      SYMB=SYM(N)
      YDIFFR=YMAX-Y(N)
      IF(YDIFFR)300,310,310
  300 YDIFFR=0.0
  310 K=101.0-(100.0*YDIFFR)/YR+.5
  320 CALL FORM2(SYMB,XY(L,K))
      IF(MOD(IC,5))340,330,340
  330 W=TP
      GO TO 350
  340 W=TC
  350 WRITE (6,8010)X,W,(XY(1,N),N=1,101),W,X
      IC=IC+1
      GO TO 440
  360 M=6-JX
      LL=50+M
      T=JX
      IF(5-JX)370,370,380
  370 T=0.0
  380 RLAB=XMAX-(T*XIJ)/5.0
      W=TC
      K=52
      DO 410 L=M,LL
      K=K-1
      I=MOD(L,5)
      IF(I-1)400,390,400
  390 W=TP
      WRITE (6,8010)RLAB,W,(XY(K,N),N=1,101),W,RLAB
      RLAB=RLAB-XIJ
      W=TC
      GO TO 410
  400 WRITE (6,8020)W,(XY(K,N),N=1,101),W
  410 CONTINUE
  420 KL=1
      GO TO 20
  430 NCC=1
  440 RETURN
      END
      SUBROUTINE SCALE(YMIN,YMAX,YINT,JY,TYMIN,TYMAX,YIJ)
C
C     SUBROUTINE 'SCALE' CALCULATES THE SCALING FOR 'PLOTR'
C
      DIMENSION C(10)
      DATA C        /1.0,1.5,2.0,3.0,4.0,5.0,7.5,10.0,15.0,20.0/
        DATA TEST / 0.76293945E-05/
   10 YR=YMAX-YMIN
      TT=YR/YINT
      IF (TT.LE.0.0) TT = 1.0
      J = ALOG10(TT)+TEST
      E=10.0**J
      TT=TT/E
      I=0
      IF(TT-1.0+TEST)20,30,30
   20 TT=TT*10.0
      E=E/10.0
   30 I=I+1
      IF(9-I)40,50,50
   40 E=E*10.0
      I=1
   50 IF(TT-C(I))60,70,30
   60 YIJ=C(I)*E
      GO TO 90
   70 Y=YMIN/C(I)
      J=Y
      T=J
      IF(0.0001-ABS(T-Y))80,60,60
   80 YIJ=C(I+1)*E
   90 X=((YMAX+YMIN)/YIJ-YINT )/2.0+.00001
      K=X
      IF(K)100,120,120
  100 Y=K
      IF(X-Y)110,120,110
  110 K=K-1
  120 TYMIN=K
      TYMIN=YIJ*TYMIN
      TYMAX=TYMIN+YINT*YIJ
```

FILE. . .   CL0516   FORTRAN P1

```
      IF (YMAX-TYMAX-TEST)130,130,30                                    CAT09370
  130 YIJJ=C(I)*E                                                       CAT09380
      XT=((YMAX+YMIN)/YIJJ-YINT)/2.0+.00001                             CAT09390
      KT=XT                                                             CAT09400
      IF (KT) 140,150,150                                               CAT09410
  140 YT=KT                                                             CAT09420
      IF (XT.NE.YT) KT=KT-1                                             CAT09430
  150 TYMINT=KT                                                         CAT09440
      TYMINT=YIJJ*TYMINT                                                CAT09450
      TYMAXT=TYMINT+YINT*YIJJ                                           CAT09460
      IF (YMAX-TYMAXT.GT.TEST) GO TO 160                                CAT09470
      TYMIN=TYMINT                                                      CAT09480
      TYMAX=TYMAXT                                                      CAT09490
      YIJ=YIJJ                                                          CAT09500
      K=KT                                                              CAT09510
  160 TT=YINT/10.0                                                      CAT09520
      JY=TT+.000001                                                     CAT09530
      YIJ=YINT*(YIJ/10.0)                                               CAT09540
      J=TYMIN/ YIJ                                                      CAT09550
      IF (K)170,180,180                                                 CAT09560
  170 J=J-1                                                             CAT09570
  180 J=J*JY+JY-K                                                       CAT09580
      JY=J                                                              CAT09590
      RETURN                                                            CAT09600
      END                                                               CAT09610
      SUBROUTINE FORM2(SYMB,XY)                                         CAT09620
      INTEGER XY,SYMB,BLANK/'  '/,TEST(18)/'2 ','3 ','4 ','5 ','6 ',   CAT09630
     1'7 ','8 ','9 ','A ','B ','C ','D ','E ','F ','G ','H ','I ','/ '/ CAT09640
      IF(XY.EQ.BLANK)GO TO 20                                           CAT09650
      DO 10 I=1,17                                                      CAT09660
      IF(XY.NE.TEST(I))GO TO 10                                         CAT09670
C         PUT IN NEXT SYMBOL OF ARRAY FOR MULTIPLE POINTS               CAT09680
      XY=TEST(I+1)                                                      CAT09690
      GO TO 30                                                          CAT09700
   10 CONTINUE                                                          CAT09710
      IF(XY.EQ.TEST(18))GO TO 30                                        CAT09720
C         IF OTHER THAN CHARACTERS IN ARRAY TEST PUT IN CHARACTER 2.    CAT09730
      XY=TEST(1)                                                        CAT09740
      GO TO 30                                                          CAT09750
C         IF BLANK, PUT IN SYMBOL                                       CAT09760
   20 XY=SYMB                                                           CAT09770
   30 RETURN                                                            CAT09780
      END                                                               CAT09790
```

APPENDIX B

SAMPLE INPUT DATA

APPENDIX B

SAMPLE INPUT DATA

Sample input data are as follows:

```
FILE. . .      H2OTEST  DATA      P1

PROBLM         H2O   20   05   00   00   05   01   05.1000  1 1   01      01  20000
LABELS0001  NUMB0002    H0003     C0004     O0005     Y
(F2.0,F4.0,2F3.0,F6.0)
01    5 10  3   2.5
02    4 10  4   2.0
03    5  9  2   2.0
04    6  8  5   3.0
05    5  9  5   2.5
06    6 10  3   3.0
07    4  8  5   2.0
08    7 10  4   3.5
09    5  9  1   1.0
10    6  9  4   3.0
11    7  9  3   3.0
12    4  8  2   2.0
13    5  7  4   2.5
14    7  7  5   3.5
15    7  7  2   2.0
16    6  8  1   1.0
17    4  7  3   2.0
18    7  7  1   1.0
19    6  8  2   2.0
20    4 10  1   1.0
CONDEL
FINISH
```

APPENDIX C

SAMPLE OUTPUT DATA

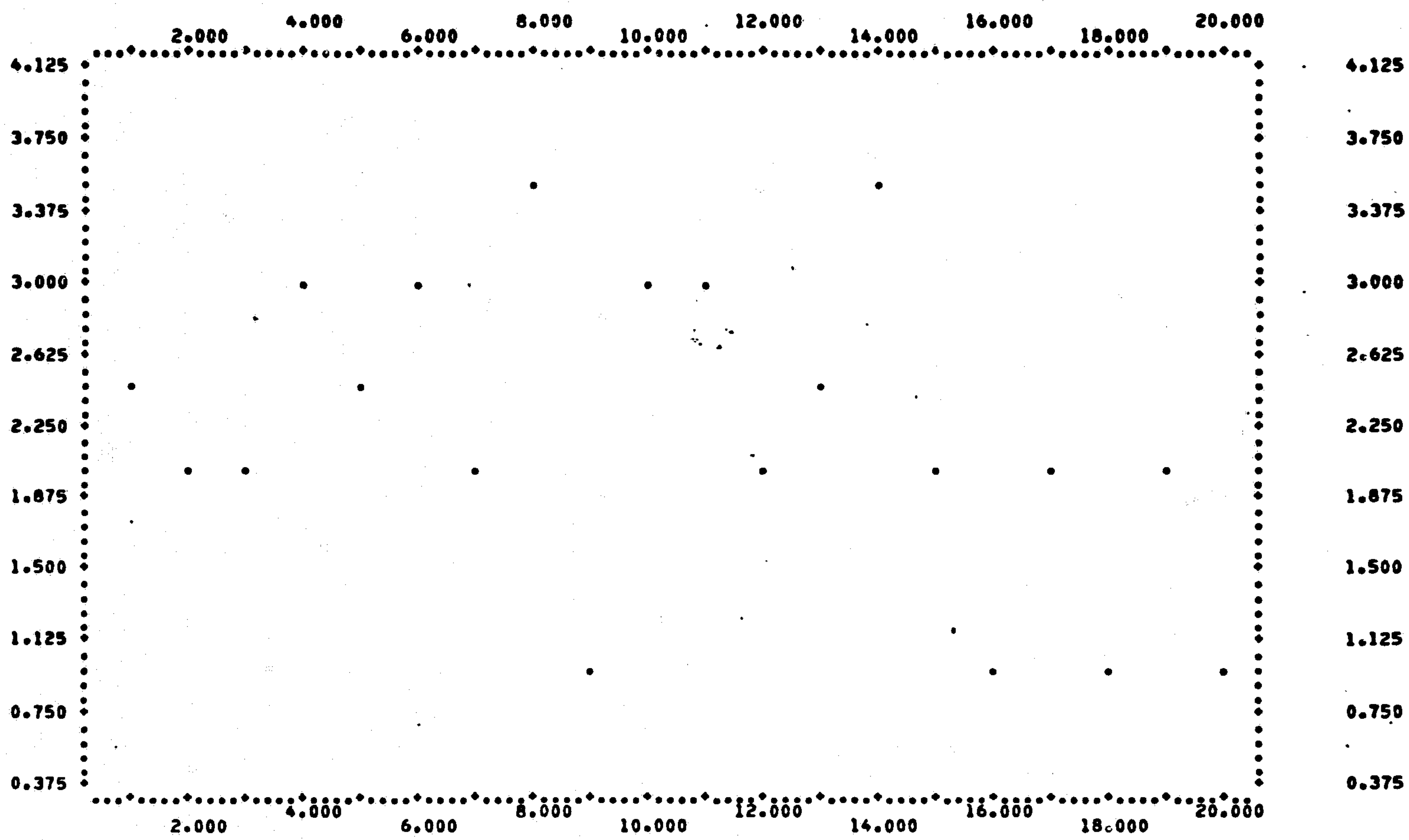
APPENDIX C

SAMPLE OUTPUT DATA

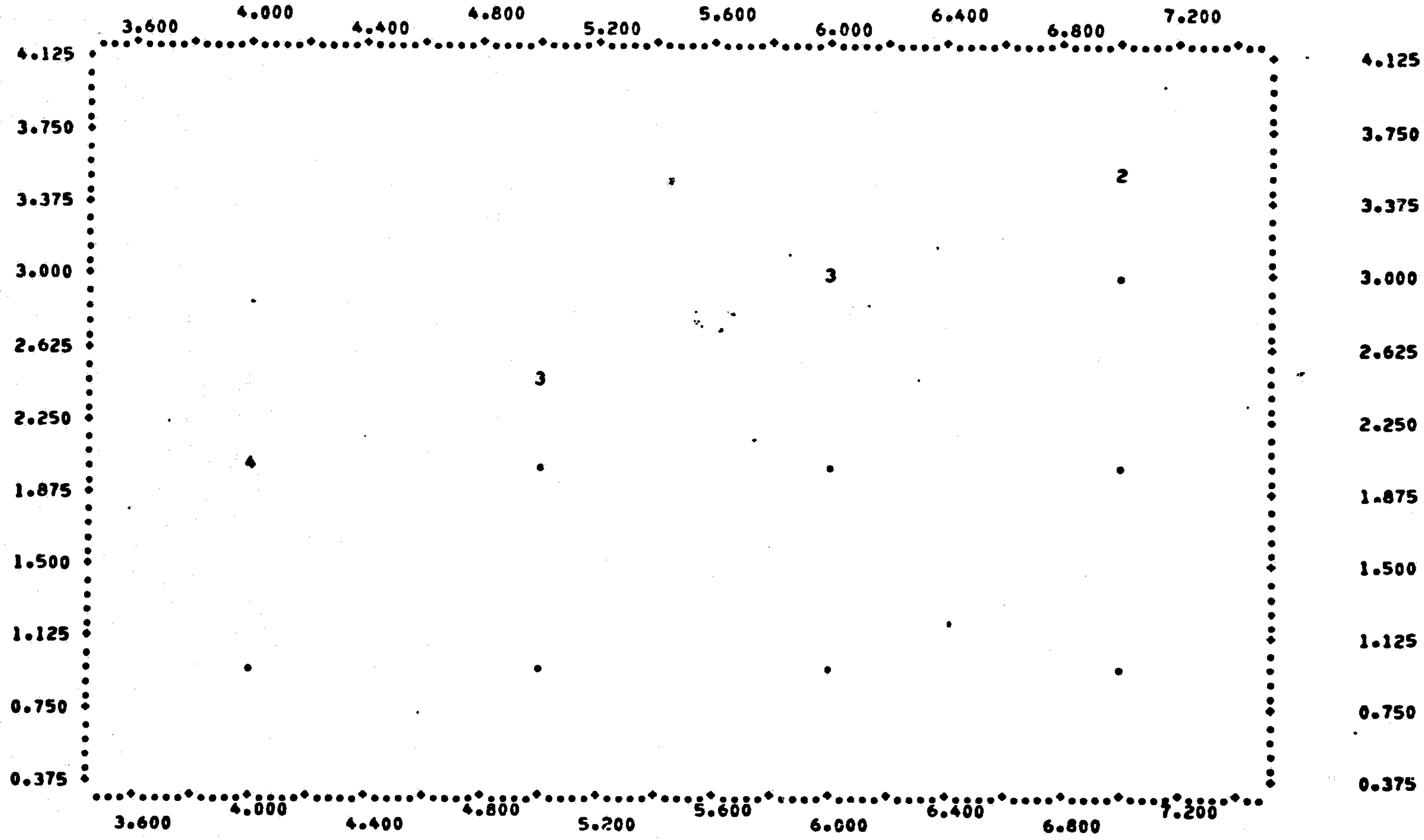
Sample output data are provided on the following pages.

```
CATE LIEBIG CRITICAL LEVEL ANALYSIS

PROGRAM CODE                          H20
NUMBER OF CASES                        20
NUMBER OF ORIGINAL VARIABLES            5
INDEX OF DEPENDENT VARIABLE             5
NUMBER OF VARIABLE FORMAT CARDS         1

NUMBER OF VARIABLES ADDED BY TRANSGENERATION  0  TOTAL NUMBER OF VARIABLES   5
VARIABLES ARE LABELED
CONTROL DELETE CARDS USED
PLOT RAW DATA
PRINT AND PLOT EACH ALLOCATION
PRINT SUMMARY OF SPLIT FOR EACH ITERATION
ALLOCATIONS LIMITED TO VARIABLES WITH AT LEAST    2  OBSERVATIONS
ALLOCATIONS LIMITED TO VARIABLES WITH A RSQ GREATER THAN 0.10
```

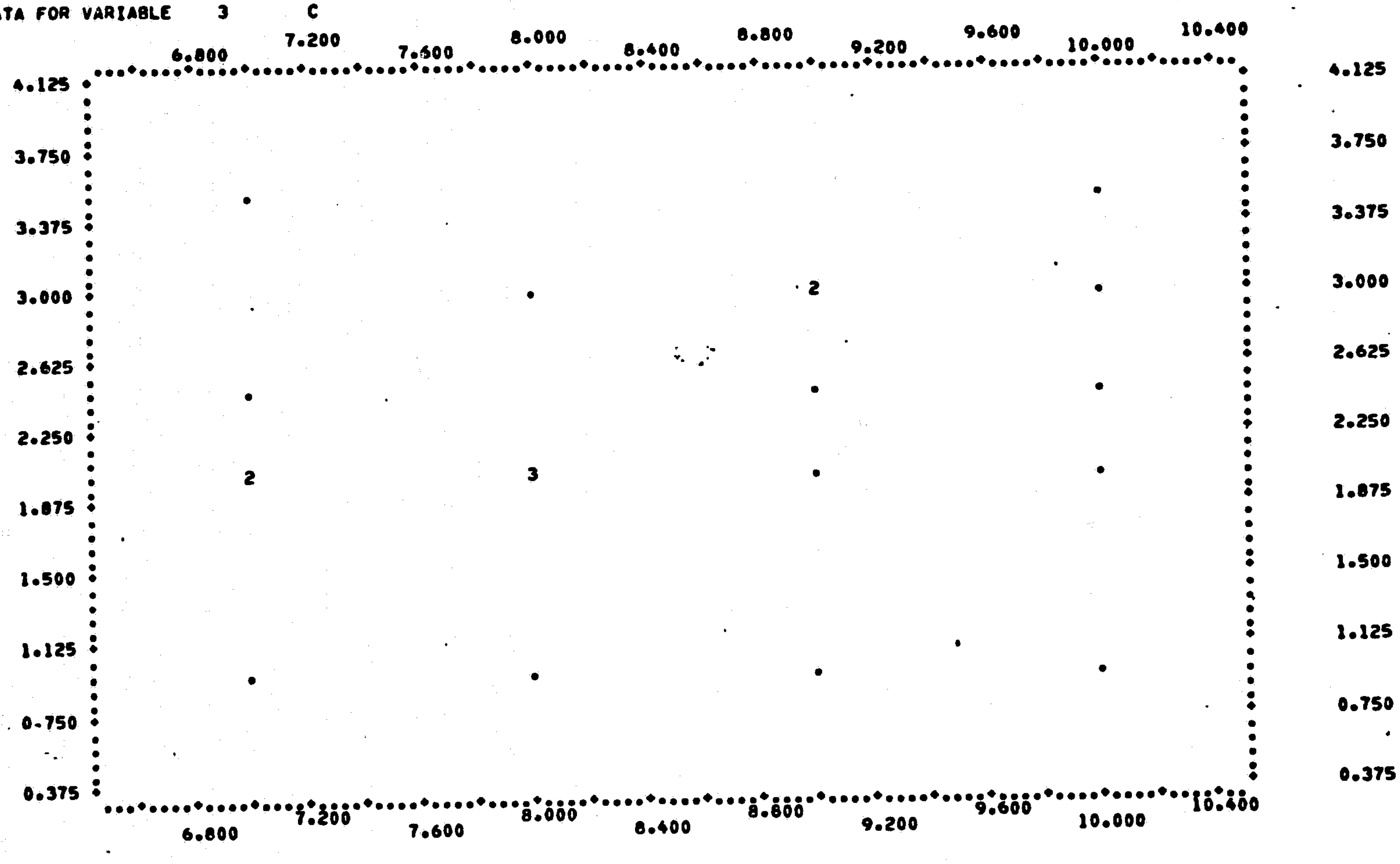
RAW DATA FOR VARIABLE 1 NUMB



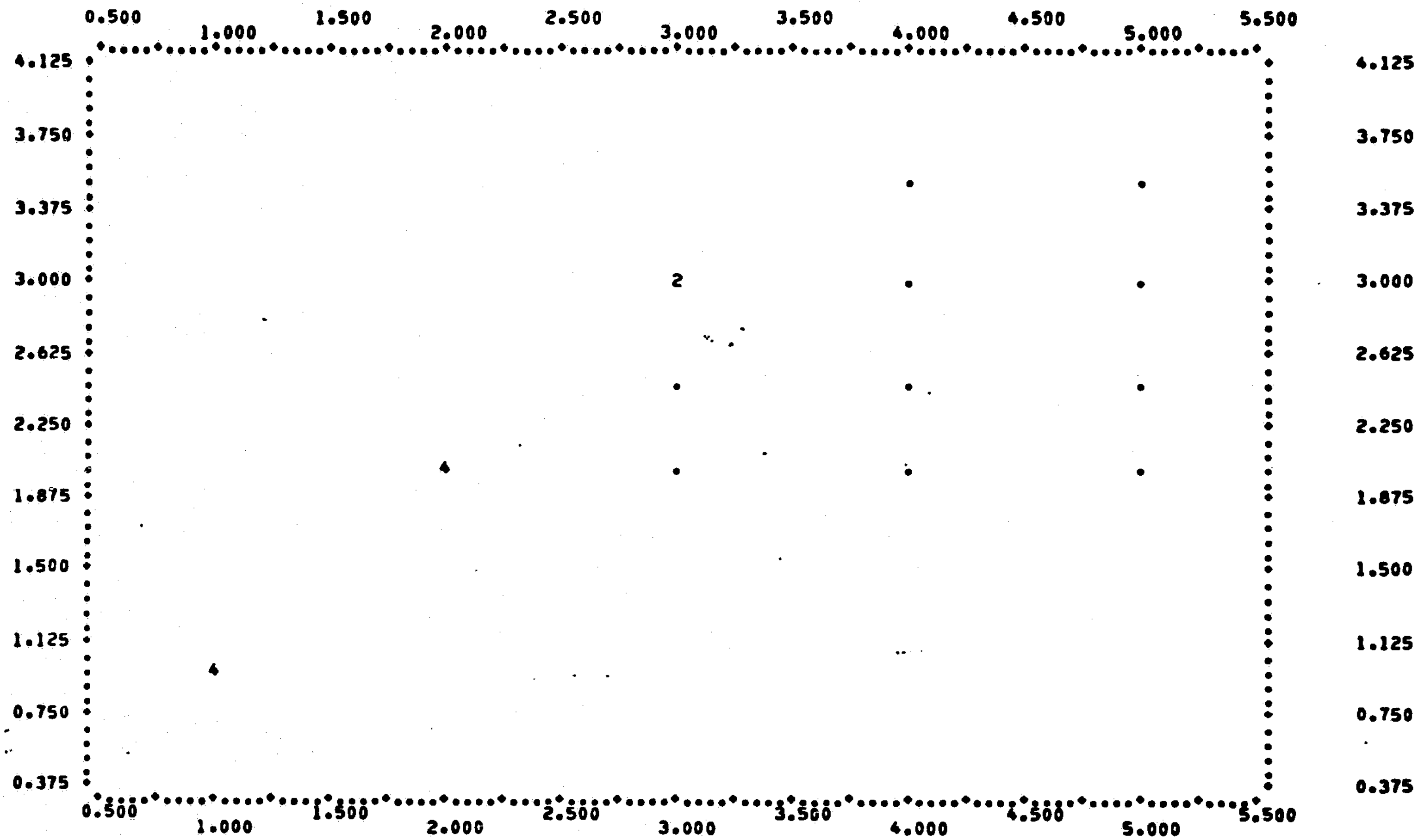
RAW DATA FOR VARIABLE 2 H



RAW DATA FOR VARIABLE 3 C



RAW DATA FOR VARIABLE 4 0



SUMMARY OF SPLITS

	VARIABLE	SIGN	CRITICAL LEVEL	0 MEAN X	Y	NUMBER CASES	1 MEAN	NUMBER CASES	RSQ
NUMB	1	-1.	15.00	18.00	1.40	5.	2.50	15.	0.37
H	2	1.	5.00	4.50	1.95	10.	2.50	10.	0.12
C	3	1.	8.00	7.50	2.10	10.	2.35	10.	0.03
O	4	1.	1.00	1.00	1.00	4.	2.53	16.	0.61

SUMMARY OF SPLITS

NUMB	VARIABLE	SIGN	CRITICAL LEVEL	0 MEAN X	Y	NUMBER CASES	1 MEAN	NUMBER CASES	RSQ
	1	-1.	14.00	17.00	2.00	3.	2.65	13.	0.22
H	2	1.	5.00	4.50	2.19	8.	2.88	8.	0.40
C	3	1.	8.00	7.50	2.38	8.	2.69	8.	0.08
O	4	1.	2.00	2.00	2.00	4.	2.71	12.	0.32

ORIGINAL PAGE IS
OF POOR QUALITY

SUMMARY OF SPLITS

	VARIABLE	SIGN	CRITICAL LEVEL	0 MEAN X	Y	NUMBER CASES	1 MEAN	NUMBER CASES	RSQ
NUMB	1	-1.	14.00	17.00	2.00	1.	2.77	11.	0.17
H	2	1.	5.00	4.50	2.25	6.	3.17	6.	0.78
C	3	1.	8.00	7.40	2.60	5.	2.79	7.	0.03
O	4	1.	3.00	3.00	2.63	4.	2.75	8.	0.01

SUMMARY OF SPLITS

	VARIABLE	SIGN	CRITICAL LEVEL	0 MEAN X	Y	NUMBER CASES	1 MEAN	NUMBER CASES	RSQ
NUMB	1	1.	11.00	7.80	3.10	5.	3.50	1.	0.40
H	2	1.	6.00	6.00	3.00	3.	3.33	3.	0.50
C	3	-1.	7.00	9.20	3.10	5.	3.50	1.	0.40
O	4	1.	3.00	3.00	3.00	2.	3.25	4.	0.25

SUMMARY OF SPLITS

	VARIABLE	SIGN	CRITICAL LEVEL	0 MEAN X	Y	NUMBER CASES	1 MEAN	NUMBER CASES	RSQ
NUMB	1	1.	10.00	7.33	3.17	3.	3.50	1.	0.33
H	2	1.	6.00	6.00	3.00	2.	3.50	2.	1.00
C	3	1.	9.00	8.00	3.17	3.	3.50	1.	0.33
O	4	-1.	4.00	5.00	3.25	2.	3.25	2.	0.0

SUMMARY OF SPLITS

	VARIABLE	SIGN	CRITICAL LEVEL	0 MEAN X	Y	NUMBER CASES	1 MEAN	NUMBER CASES	RSQ
NUMB	1	-1.	8.00	14.00	3.50	1.	3.50	1.	0.0
H	2	0.	0.0	0.0	0.0	0.	0.0	0.	0.0
C	3	-1.	7.00	10.00	3.50	1.	3.50	1.	0.0
O	4	-1.	4.00	5.00	3.50	1.	3.50	1.	0.0

INITIAL ALLOCATIONS

	VARIABLE	SIGN	CRITICAL LEVEL	0 MEAN X	Y	NUMBER CASES	1 MEAN	NUMBER CASES	RSQ
O	4	1.	1.00	1.00	1.00	4.	2.53	16.	0.61
O	4	1.	2.00	2.00	2.00	4.	2.71	12.	0.32
H	2	1.	5.00	4.50	2.25	6.	3.17	6.	0.78
O	4	1.	3.00	3.00	3.00	2.	3.25	4.	0.25
H	2	1.	6.00	6.00	3.00	2.	3.50	2.	1.00

VARIABLES WITH LESS THAN 2 CASES WERE NOT ALLOCATED.
VARIABLES WITH SPLITS HAVING A RSQ LESS THAN 0.10 WERE NOT ALLOCATED.

CONSISTENCY CHECK

CASE	ORIGINAL	NEW	ACTUAL Y	ORIGINAL LIMITING X	ORIGINAL EST Y	NEW LIMITING X	NEW EST Y
12	4	2	2.00000	2.00000	2.00000	4.00000	1.86539

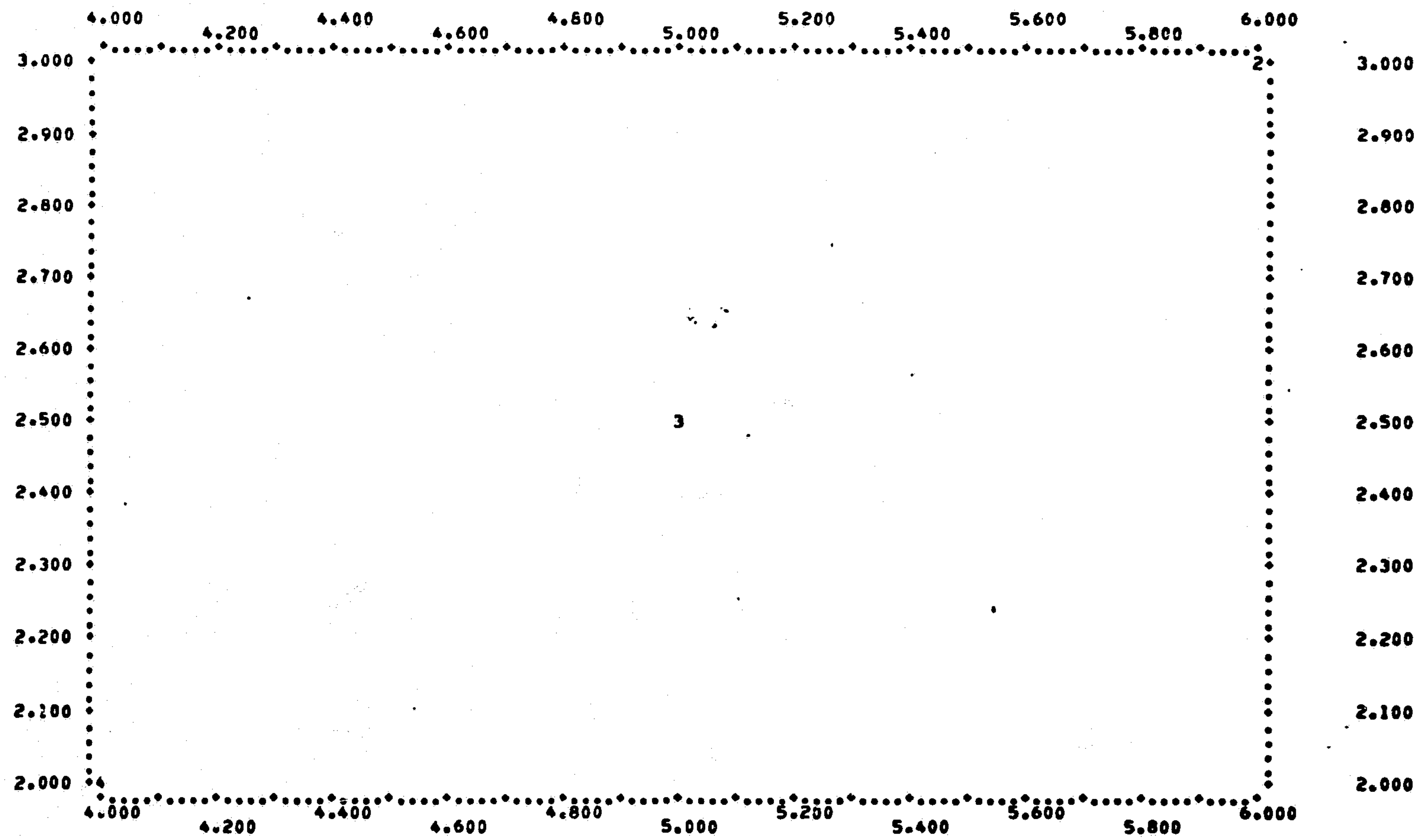
FINAL ALLOCATIONS

	VARIABLE	SIGN	CRITICAL LEVEL	0 MEAN X	Y	NUMBER CASES	1 MEAN	NUMBER CASES	RSQ
H	2	1.	6.00	4.78	2.39	9.	3.50		
O	4	1.	3.00	1.78	1.78	9.	3.50		

VARIABLE ALLOCATED 2 H 9. CASES

CASE	X	Y
1	5.0000	2.5000
2	4.0000	2.0000
4	6.0000	3.0000
5	5.0000	2.5000
7	4.0000	2.0000
10	6.0000	3.0000
12	4.0000	2.0000
13	5.0000	2.5000
17	4.0000	2.0000

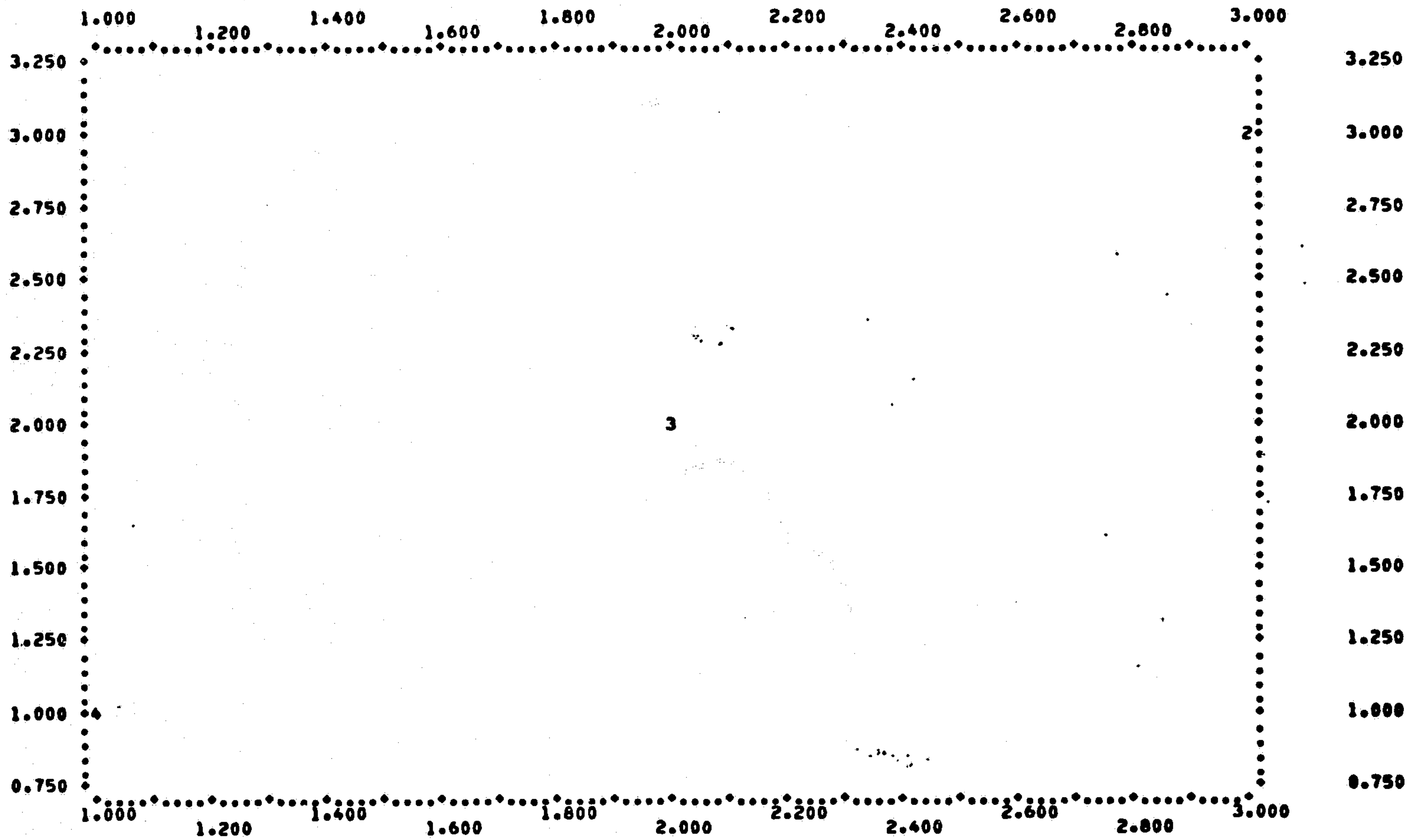
VARIABLE ALLOCATED 2 H 9. CASES



VARIABLE ALLOCATED 4 0 9. CASES

CASE	X	Y
3	2.0000	2.0000
6	3.0000	3.0000
9	1.0000	1.0000
11	3.0000	3.0000
15	2.0000	2.0000
16	1.0000	1.0000
18	1.0000	1.0000
19	2.0000	2.0000
20	1.0000	1.0000

VARIABLE ALLOCATED 4 0 9. CASES



MODEL SUMMARY

VARIABLE	NUM	INTERCEPT	SLOPE	CRITICAL LEVEL
H	2	-0.69355	0.64516	6.00000
O	4	0.0	1.00000	3.00000

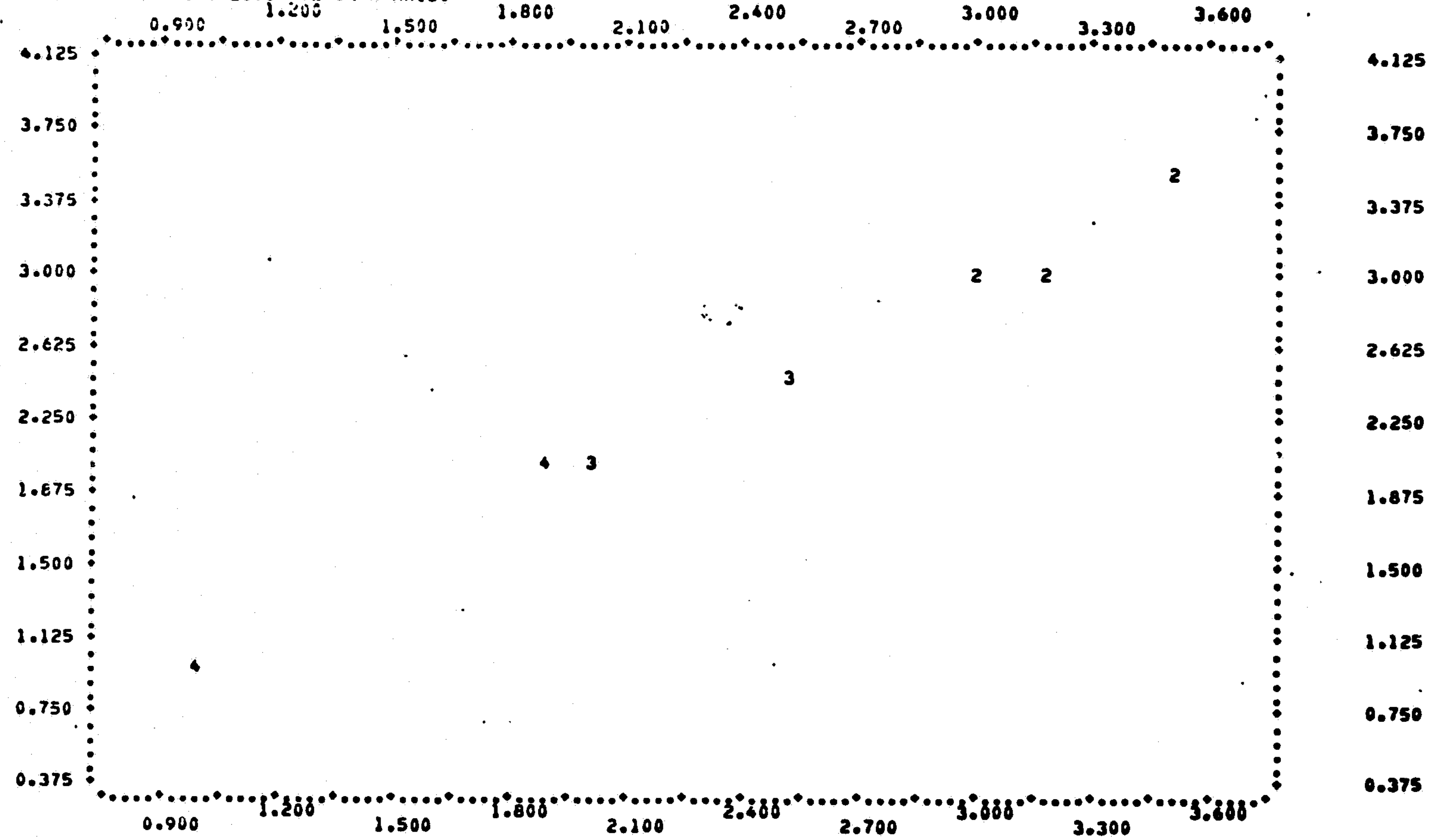
MAXIMUM 3.50000

DEPENDENT DATA

CASE	VARIABLE	LIMITING X	EST Y	ACT Y	ERROR
1	2	5.0000	2.5323	2.5000	-0.0323
2	2	4.0000	1.8871	2.0000	0.1129
3	4	2.0000	2.0000	2.0000	0.0
4	2	6.0000	3.1774	3.0000	-0.1774
5	2	5.0000	2.5323	2.5000	-0.0323
6	4	3.0000	3.0000	3.0000	0.0
7	2	4.0000	1.8871	2.0000	0.1129
8	4	4.0000	3.5000	3.5000	0.0
9	4	1.0000	1.0000	1.0000	0.0
10	2	6.0000	3.1774	3.0000	-0.1774
11	4	3.0000	3.0000	3.0000	0.0
12	2	4.0000	1.8871	2.0000	0.1129
13	2	5.0000	2.5323	2.5000	-0.0323
14	4	5.0000	3.5000	3.5000	0.0
15	4	2.0000	2.0000	2.0000	0.0
16	4	1.0000	1.0000	1.0000	0.0
17	2	4.0000	1.8871	2.0000	0.1129
18	4	1.0000	1.0000	1.0000	0.0
19	4	2.0000	2.0000	2.0000	0.0
20	4	1.0000	1.0000	1.0000	0.0

MEAN ERROR	0.00
ROOT MEAN SQUARE ERROR	0.08
MEAN SQUARE ERROR	0.01
RSQ	0.99

ACTUAL Y ON Y-AXIS, ESTIMATE ON X-AXIS.



FINISH CARD ENCOUNTERED
PROGRAM TERMINATED